

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.:	09/466,275	§	
Filed:	December 17, 1999	§	Examiner: Chavis, John Q.
Inventor(s):		§	Group/Art Unit: 2124
	Murali Parthasarathy and Omid	§	Atty. Dkt. No: 5150-18302
	Sojoodi	§	
		§	
		§	
Title:	System and Method for	§	
	Creating a Graphical	§	
	Program Which	§	
	Invokes Methods and	§	
	Properties of Objects	§	
		§	
		§	

**RE-ASSERTED PETITION
TO WITHDRAW HOLDING OF ABANDONMENT
UNDER CFR 1.181(a)**

Attention: Office of Petitions
Mail Stop Petition
Commissioner for Patents
P.O. Box 1450 Alexandria, VA 22313-1450
FAX (571) 273-8300

Applicant hereby re-asserts the 37 CFR 1.181 Petition in response to the Notice of Abandonment in the above-identified application as well as lack of communication from the Patent Office regarding Applicant's earlier 37 CFR 1.181 Petition and Status Inquiry.

NOTICE OF ABANDONMENT

The Notice indicates Applicant did not submit a timely and proper reply within the statutory period of one month from the date of an Office Action (Notice of Non-Compliant Amendment) mailed on January 23, 2004; specifically, the notice indicates

that the certificate of mailing was not signed and dated and that the Response was not signed.

Applicant submits that the Response mailed on February 18, 2004 was in fact signed, and the certificate of mailing was also signed. Specifically, Applicant mailed a “Response to Notice of Non-Compliant Amendment Mailed January 23, 2004-Response to Office Action of September 10, 2002” on February 18, 2004. Attached are a true and correct copy of the signed and dated “Response to Notice of Non-Compliant Amendment Mailed January 23, 2004-Response to Office Action of September 10, 2002” and a copy of the return receipt postcard stamped as received by the U.S. Patent Office on February 23, 2004, which evidences the previous submission of the Response.

Applicant respectfully requests entry of this Response in the application and requests that the Notice of Abandonment be vacated or withdrawn.

PETITION TO WITHDRAW HOLDING OF ABANDONMENT

Applicant submits that a Petition to Withdraw Holding of Abandonment under CFR 1.181(a) was mailed to the USPTO on August 12, 2005 with a signed certificate of mailing. The 1.181 Petition asked the Commissioner of Patents to vacate or withdraw the Notice of Abandonment. Applicant submits that the 1.181 Petition was mailed with a true and correct copy of the signed and dated “Response to Notice of Non-Compliant Amendment Mailed January 23, 2004-Response to Office Action of September 10, 2002” and a copy of the return receipt postcard stamped as received by the U.S. Patent Office on February 23, 2004, which evidences the previous submission of the Response. Applicant encloses a copy of the return receipt postcard stamped as received by the U.S. Patent Office on August 15, 2005, which evidences the previous submission of the 1.181 Petition.

However, Applicant did not receive any substantive correspondence back from the U.S. Patent Office regarding the 1.181 Petition.

Applicant respectfully requests entry of this 1.181 Petition in the application and requests that the Notice of Abandonment be vacated or withdrawn.

STATUS INQUIRY

Applicant submits that a Status Inquiry was mailed to the USPTO on 1/29/07 with an electronic signature. The Status Inquiry was directed to the Patent Office notifying the Applicant as to the status of this matter. Applicant encloses a copy of the electronic acknowledgement receipt which indicates the U.S. Patent Office received the Status Inquiry on 1/29/07.

However, Applicant did not receive any correspondence back from the U.S. Patent Office regarding the Status Inquiry.

CONCLUSION

Applicant submits that the Response to the Office Action of January 23, 2004 and the Petition to Withdraw Holding of Abandonment under CFR 1.181(a) of August 12, 2005 were properly and timely filed at the USPTO. Thus Applicant re-asserts the earlier 1.181 Petition and respectfully requests that the Notice of Abandonment be vacated or withdrawn.

No fees are believed necessary; however, should any fees be required, please appropriately charge those fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-18302/JCH.

Respectfully submitted,

/Jeffrey C. Hood/
Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANTS

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: 2007-11-21 JCH/MRW

IN THE SPECIFICATION:

Please amend the Abstract as follows:

A method for developing an automation client program in a graphical programming environment is disclosed. The graphical programming environment provides a set of automation nodes and controls which may be dropped and wired together to create a graphical program. The nodes include an automation refnum which references a user-selected automation class from an automation type library ~~exported by an automation server application, such as Microsoft Excel;~~ an automation open node which instantiates an object from the selected automation class; an automation invoke node which invokes a user-selected method of the automation class; and an automation property node which invokes, i.e., reads or writes, user-selected properties of the automation class. The nodes enable the displaying, manipulating, cataloging, editing or performance other operations, such as may be performed by an automation server, on data acquired or generated by a virtual instrument. A method for performing class propagation and type propagation checking of automation objects in a graphical program is also disclosed. ~~The automation class of a first automation node is propagated from the first node to a second automation node when the two nodes are wired together or when the automation class of the first node is changed to a second class. The automation invoke node and automation property node perform type checking to verify that the user-selected method or property is valid for, i.e., defined by, the automation class of the node. The node requests an object manager to determine whether or not the method or property is valid. The object manager queries a type library which the automation class is in, in order to obtain a list of valid methods and properties for the automation class. The object manager searches the list to determine if the specified method or property is present in the list, i.e., is valid.~~

IN THE CLAIMS:

Please amend the claims as follows:

1. (Original) A computer-implemented method for creating a graphical data flow program, wherein the graphical data flow program is operable to invoke a method of an object, wherein the method for creating the graphical data flow program operates in a computer including a display and a user input device, the method for creating the graphical data flow program comprising:

displaying on the screen a node in the graphical data flow program in response to user input, wherein the node is operable to invoke a method of an object;

configuring the node to receive information on the object in response to user input, wherein said configuring comprises connecting the information on the object to an input of the node;

wherein, during execution of the graphical data flow program, the node is operable to invoke the method of the object.

2. (Original) The computer-implemented method of claim 1, wherein the node includes an object reference input for receiving a reference to the object;

wherein said configuring comprises connecting said object reference input of the node to receive the reference to the object;

wherein the node receives the information on the object on the object reference input during execution of the graphical data flow program.

3. (Original) The computer-implemented method of claim 2, wherein said configuring comprises:

displaying on the screen an object reference node which includes an object reference output that provides the reference to the object; and

connecting the object reference output of the object reference node to the object reference input of the node.

4. (Currently Amended) The computer-implemented method of claim 3, further comprising:

executing the graphical data flow program, wherein said executing includes propagating the reference to the object from the object reference output of the object reference node to the object reference input of the node.

5. (Original) The computer-implemented method of claim 1, further comprising:

executing the graphical data flow program, wherein said executing includes propagating the information on the object to the node.

6. (Currently Amended) The computer-implemented method of claim 1, wherein the object is comprised in a server, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input;

displaying on the screen a list of possible classes from the selected library;

selecting a class from the list of possible classes in response to user input;

wherein the object is instantiated from the class.

7. (Original) The computer-implemented method of claim 1, further comprising:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the method of the object; and

executing said execution instructions, wherein the node invokes the method of the object during said executing.

8. (Original) The computer-implemented method of claim 7, wherein said executing includes propagating the information on the object to the node.

9. (Original) The computer-implemented method of claim 1, wherein the graphical data flow program is operable to invoke the method of the object for performing instrumentation functions on an instrument.

10. (Original) The computer-implemented method of claim 1, wherein the graphical data flow program includes a block diagram and a front panel, wherein the block diagram includes the node.

11. (Original) A computer-implemented method for creating a graphical data flow program, wherein the graphical data flow program is operable to invoke a property of an object, wherein the method for creating the graphical data flow program operates in a computer including a display screen and a user input device, the method for creating the graphical data flow program comprising:

displaying on the screen a node in the graphical data flow program in response to user input, wherein the node is operable to invoke a property of an object;

configuring the node to receive information on the object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to invoke the property of the object.

12. (Original) The computer-implemented method of claim 11, wherein the node includes an object reference input for receiving a reference to the object;

wherein said configuring comprises connecting the object reference input of the node to receive the reference to the object;

wherein the node receives the information on the object on the object reference input during execution of the graphical data flow program.

13. (Original) The computer-implemented method of claim 12, wherein said configuring comprises:

displaying on the screen an object reference node which includes an object reference output that provides the reference to the object; and

connecting the object reference output of the object reference node to the object reference input of the node.

14. (Currently Amended) The computer-implemented method of claim ~~11~~ 13, further comprising:

executing the graphical data flow program, wherein said executing includes propagating the reference to the object from the object reference output of the object reference node to the object reference input of the node.

15. (Currently Amended) The computer-implemented method of claim 11, further comprising:

executing the graphical data flow program, wherein said executing includes ~~propagating~~ providing the information on the object to the node.

16. (Original) The computer-implemented method of claim 11, wherein the object is comprised in a server, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input

displaying on the screen a list of possible classes from the selected library;

selecting a class from the list of possible classes in response to user input;

wherein the object is instantiated from the class.

17. (Original) The computer-implemented method of claim 11, further comprising:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the property of the object; and

executing said execution instructions, wherein the node invokes the property of the object during said executing.

18. (Original) The computer-implemented method of claim 11, wherein the node is operable to get and/or set one or more properties of the object.

19. (Original) The computer-implemented method of claim 11, wherein the graphical data flow program is operable to invoke the property of the object for performing instrumentation functions on an instrument.

20. (Original) The computer-implemented method of claim 11, wherein the graphical data flow program includes a block diagram and a front panel, wherein the block diagram includes the node.

21. (Original) A memory medium comprising program instructions for creating a graphical data flow program, wherein the graphical data flow program is operable to invoke a method of an object, wherein the program instructions are executable to:

display on the screen a node in the graphical data flow program in response to user input, wherein the node is operable to invoke a method of an object;

configure the node to receive information on the object in response to user input, wherein said configuring comprises connecting the information on the object to an input of the node;

wherein, during execution of the graphical data flow program, the node is operable to invoke the method of the object.

22. (Original) The memory medium of claim 21, wherein the node includes an object reference input for receiving a reference to the object;

wherein said configuring comprises connecting the object reference input of the node to receive the reference to the object;

wherein the node receives the information on the object on the object reference input during execution of the graphical data flow program.

23. (Original) The memory medium of claim 22, wherein said configuring comprises:

displaying on the screen an object reference node which includes an object reference output that provides the reference to the object; and

connecting the object reference output of the object reference node to the object reference input of the node.

24. (Original) The memory medium of claim 23, wherein the program instructions are further executable to:

execute the graphical data flow program, wherein said executing includes propagating the reference to the object from the object reference output of the object reference node to the object reference input of the node

25. (Currently Amended) The ~~computer-implemented method~~ memory medium of claim ~~[[1]]~~ 21, wherein the program instructions are further executable to:

construct execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the method of the object; and

execute said execution instructions, wherein the node invokes the method of the object during said executing.

26. (Original) The memory medium of claim 25, wherein said executing includes propagating the information on the object to the node.

27. (Original) The memory medium of claim 21, wherein the graphical data flow program is operable to invoke the method of the object for performing instrumentation functions on an instrument.

28. (Original) The memory medium of claim 21, wherein the graphical data flow program includes a block diagram and a front panel, wherein the block diagram includes the node.

29. (Original) A memory medium comprising program instructions for creating a graphical data flow program, wherein the graphical data flow program is operable to invoke a property of an object, wherein the program instructions are executable to:

display on the screen a node in the graphical data flow program in response to user input, wherein the node is operable to invoke a property of an object;

configure the node to receive information on the object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to invoke the property of the object.

30. (Original) The memory medium of claim 29, wherein the node includes an object reference input for receiving a reference to the object;

wherein said configuring comprises connecting the object reference input of the node to receive the reference to the object;

wherein the node receives the information on the object on the object reference input during execution of the graphical data flow program.

31. (Original) The memory medium of claim 30, wherein said configuring comprises:

displaying on the screen an object reference node which includes an object reference output that provides the reference to the object; and

connecting the object reference output of the object reference node to the object reference input of the node.

32. (Currently Amended) The memory medium of claim ~~29~~ 31, wherein the program instructions are further executable to:

execute the graphical data flow program, wherein said executing includes propagating the reference to the object from the object reference output of the object reference node to the object reference input of the node.

33. (Original) The memory medium of claim 29, wherein the program instructions are further executable to:

construct execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the property of the object; and

execute said execution instructions, wherein the node invokes the property of the object during said executing.

34. (Original) The memory medium of claim 29, wherein the node is operable to get and/or set one or more properties of the object.

35. (Original) The memory medium of claim 29, wherein the graphical data flow program is operable to invoke the property of the object for performing instrumentation functions on an instrument.

36. (Original) The memory medium of claim 29, wherein the graphical data flow program includes a block diagram and a front panel, wherein the block diagram includes the node.

37. (Original) A memory medium which stores a graphical data flow program; wherein the graphical data flow program includes a node which is operable to invoke a method of an object;

wherein the node includes an input which is configurable to receive information on the object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to invoke the method of the object.

38. (Original) A memory medium which stores a graphical data flow program; wherein the graphical data flow program includes a node which is operable to invoke a property of an object;

wherein the node includes an input which is configurable to receive information on the object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to invoke the property of the object.

39. (New) A computer-implemented method for creating a graphical data flow program, the method for creating the graphical data flow program comprising:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to invoke a method of any of a plurality of software objects;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to invoke a method of the first object.

40. (New) The computer-implemented method of claim 39,

wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

41. (New) The computer-implemented method of claim 39,

wherein said configuring comprises connecting the information on the first object to an input of the node.

42. (New) The computer-implemented method of claim 39, wherein said configuring comprises:

- displaying on the screen a list of libraries associated with one or more servers;
- selecting a library from the list of libraries in response to user input;
- displaying on the screen a list of possible classes from the selected library;
- selecting a class from the list of possible classes in response to user input;
- wherein the first object is instantiated from the class.

43. (New) The computer-implemented method of claim 39, further comprising:
constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the method of the first object; and

executing said execution instructions, wherein the node invokes the method of the first object during said executing.

44. (New) The computer-implemented method of claim 39, further comprising:
configuring the node to receive information on a first method of the first object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to invoke the first method of the first object.

45. (New) The computer-implemented method of claim 39,
wherein said configuring the node to receive information on a first object comprises configuring the node to receive information on a first class of the first object, wherein the first object is instantiated from the first class.

46. (New) A memory medium comprising program instructions for creating a graphical data flow program, wherein the program instructions are executable to implement:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to invoke a method of any of a plurality of software objects;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to invoke a method of the first object.

47. (New) The memory medium of claim 46,

wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

48. (New) The memory medium of claim 46,

wherein said configuring comprises connecting the information on the first object to an input of the node.

49. (New) The memory medium of claim 46, wherein said configuring comprises:

displaying on the screen a list of possible classes;

selecting a class from the list of possible classes in response to user input;

wherein the first object is instantiated from the class.

50. (New) The memory medium of claim 46, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input;
displaying on the screen a list of possible classes from the selected library;
selecting a class from the list of possible classes in response to user input;
wherein the first object is instantiated from the class.

51. (New) The memory medium of claim 46, wherein the program instructions are further executable to implement:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the method of the first object; and

executing said execution instructions, wherein the node invokes the method of the first object during said executing.

52. (New) The memory medium of claim 46, wherein the program instructions are further executable to implement:

configuring the node to receive information on a first method of the first object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to invoke the first method of the first object.

53. (New) The memory medium of claim 46,

wherein said configuring the node to receive information on a first object comprises configuring the node to receive information on a first class of the first object, wherein the first object is instantiated from the first class.

54. (New) A computer-implemented method for creating a graphical data flow program, the method for creating the graphical data flow program comprising:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate

functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to perform at least one of getting or setting at least one property of any of a plurality of software objects;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to perform at least one of getting or setting at least one property of the first object.

55. (New) The computer-implemented method of claim 54,

wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

56. (New) The computer-implemented method of claim 54,

wherein said configuring comprises connecting the information on the first object to an input of the node.

57. (New) The computer-implemented method of claim 54, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input;

displaying on the screen a list of possible classes from the selected library;

selecting a class from the list of possible classes in response to user input;

wherein the first object is instantiated from the class.

58. (New) The computer-implemented method of claim 54, further comprising:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to perform at least one of getting or setting at least one property of the first object; and

executing said execution instructions, wherein the node performs at least one of getting or setting at least one property of the first object during said executing.

59. (New) The computer-implemented method of claim 54, further comprising:
configuring the node to receive information on a first property of the first object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to perform at least one of getting or setting at least one property of the first object.

60. (New) The computer-implemented method of claim 54,
wherein said configuring the node to receive information on a first object comprises configuring the node to receive information on a first class of the first object, wherein the first object is instantiated from the first class.

61. (New) A memory medium comprising program instructions for creating a graphical data flow program, wherein the program instructions are executable to implement:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to perform at least one of getting or setting at least one property of any of a plurality of software objects;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to perform at least one of getting or setting at least one property of the first object.

62. (New) The memory medium of claim 61,

wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

63. (New) The memory medium of claim 61,

wherein said configuring comprises connecting the information on the first object to an input of the node.

64. (New) The memory medium of claim 61, wherein said configuring comprises:

displaying on the screen a list of possible classes;
selecting a class from the list of possible classes in response to user input;
wherein the first object is instantiated from the class.

65. (New) The memory medium of claim 61, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;
selecting a library from the list of libraries in response to user input;
displaying on the screen a list of possible classes from the selected library;
selecting a class from the list of possible classes in response to user input;
wherein the first object is instantiated from the class.

66. (New) The memory medium of claim 61, wherein the program instructions are further executable to implement:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to perform at least one of getting or setting at least one property of the first object; and

executing said execution instructions, wherein the node performs at least one of getting or setting at least one property of the first object during said executing.

67. (New) The memory medium of claim 61, wherein the program instructions are further executable to implement:

configuring the node to receive information on a first method of the first object in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to perform at least one of getting or setting at least one property of the first object.

68. (New) The memory medium of claim 61,

wherein said configuring the node to receive information on a first object comprises configuring the node to receive information on a first class of the first object, wherein the first object is instantiated from the first class.

69. (New) A memory medium comprising program instructions for creating a graphical data flow program, wherein the program instructions are executable to implement:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to invoke a method of a software object instantiated from any of a plurality of software classes;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first class in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to invoke a method of a first object instantiated from the first class.

70. (New) The memory medium of claim 69,

wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

71. (New) The memory medium of claim 69,

wherein said configuring comprises connecting the information on the first class to an input of the node.

72. (New) The memory medium of claim 69, wherein said configuring comprises:

displaying on the screen a list of possible classes; and

selecting the first class from the list of possible classes in response to user input.

73. (New) The memory medium of claim 69, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input;

displaying on the screen a list of possible classes from the selected library; and

selecting the first class from the list of possible classes in response to user input.

74. (New) The memory medium of claim 69, wherein the program instructions are further executable to implement:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the method of the first object; and

executing said execution instructions, wherein the node invokes the method of the first object during said executing.

75. (New) The memory medium of claim 69, wherein the program instructions are further executable to implement:

configuring the node to receive information on a first method of the first class in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to invoke the first method of the first object instantiated from the first class.

76. (New) A memory medium comprising program instructions for creating a graphical data flow program, wherein the program instructions are executable to implement:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to perform at least one of getting or setting at least one property of a software object instantiated from any of a plurality of software classes;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first class in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to perform at least one of getting or setting at least one property of a first object instantiated from the first class.

77. (New) The memory medium of claim 76,
wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

78. (New) The memory medium of claim 76,
wherein said configuring comprises connecting the information on the first class to an input of the node.

79. (New) The memory medium of claim 76, wherein said configuring comprises:

- displaying on the screen a list of possible classes;
- selecting the first class from the list of possible classes in response to user input.

80. (New) The memory medium of claim 76, wherein said configuring comprises:

- displaying on the screen a list of libraries associated with one or more servers;
- selecting a library from the list of libraries in response to user input;
- displaying on the screen a list of possible classes from the selected library;
- selecting the first class from the list of possible classes in response to user input.

81. (New) The memory medium of claim 76, wherein the program instructions are further executable to implement:

- constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to perform at least one of getting or setting at least one property of the first object; and

- executing said execution instructions, wherein the node performs at least one of getting or setting at least one property of the first object during said executing.

82. (New) The memory medium of claim 76, wherein the program instructions are further executable to implement:

configuring the node to receive information on a first property of the first class in response to user input;

wherein, during execution of the graphical data flow program, the node is operable to perform at least one of getting or setting the first property of the first object instantiated from the first class.

REMARKS

Applicant is in receipt of the Response to Amendment mailed 1/23/2004. Applicant submits that the present amendment, which is responsive to the Office Action mailed September 10, 2002, is in compliance with 37 CFR 1.121(c).

In the Office Action mailed September 10, 2002, the abstract was objected to, claim 4 was rejected under §112, and the claims were rejected under §102 and §103.

Objection to the Abstract

The abstract was rejected to because it contained more than 150 words. Applicant has amended the abstract to be less than 150 words. Thus, Applicant submits that this objection has been overcome.

Section 112 Rejections

Claim 4 was rejected under §112 because it did not end in a period. Applicant has amended claim 4 to correct this error. Applicant has also corrected a missing semicolon in claim 6, has corrected an improper dependency in claims 14, 25 and 32, and has corrected various other minor errors.

Section 102 Rejections

Claims 1-8, 11-18, 21-26, 29-34 and 37-38 were rejected under §102(e) as being anticipated by Fowlow et al. Applicant respectfully submits that the present claims are allowable over the Fowlow reference. The Fowlow reference relates to a method for visually constructing object oriented application software that is to be installed on a distributed object system. Figure 5 of the Fowlow patent illustrates a composition design environment which is evidently used for visually constructing object oriented application software. As shown in Figure 5 of Fowlow and discussed beginning at column 11 line 1:

“The connections between parts and other parts or interfaces is made using plugs such as shown at 544, and sockets such as shown at 546. ... a socket is a representation of a service provided by an object, comprising usually an object reference that is passed by that object to another

requesting object. A plug, conversely, is a service that an object is capable of requesting and processing. As will be known to those with skill in the object programming arts, objects communicate amongst themselves by passing and operating upon object references which communication is represented schematically by drawing connections (such as connection 536) between the plug of a first object and a socket of a second object.” (See column 11 lines 1-28)

Applicant respectfully submits that the Fowlow reference does not teach the notion of a “graphical dataflow program” as recited in the present claims. A graphical data flow program comprises a program comprising a plurality of nodes or icons, wherein the nodes or icons are interconnected by lines, and wherein the lines represent the flow of data between the nodes. The Fowlow reference does not teach or suggest the concept of a graphical data flow program. Evidently, the connections between object icons described in the Fowlow reference represent object references that are used to request services as represented by a “socket”. Thus, the Fowlow reference does not teach the notion of a graphical dataflow program. Further, the Fowlow reference does not teach the notion of a node in a graphical dataflow program that is configurable or operable to invoke a method of an object or get or set a property of an object.

With respect to the step of “displaying on the screen a node in the graphical dataflow program in response to user input”, the Office Action cites column 3 lines 41-59 of Fowlow. This cited portion of Fowlow merely refers to linking plugs and sockets of icons that represent objects. Applicant submits that this does not teach or suggest a node in a graphical data flow program. With respect to the limitation of “wherein the node is operable to invoke a method of an object” the Office Action states “see in the location above the making of a selection action on one of the icons”. Applicant has reviewed this cited portion and believes that the “selection action” refers to user input or user action of selecting an icon. This cited portion does not appear to be relevant to the concept of a node being operable to invoke a method of an object.

Similar arguments apply with respect to each of the pending independent claims that are directed to invoking methods of objects and/or getting or setting properties of objects, e.g., claims 1 - 38.

Claims 9-10, 19-20, 27-28 and 35-36 were rejected under §103(a) as being unpatentable over Fowlow as applied to claim 1 above and further in view of Meyer.

Applicant respectfully submits that these claims are allowable in view of the arguments made with respect to Fowlow above. Further, Applicant notes that Meyer does not provide at least several of the missing elements from the Fowlow reference. For example, the Meyer patent is directed toward developing “a graphical control flow structure”. The Meyer patent is based on the well known Grafcet/IEC 1131 standard, which is a standard for creating control flow diagrams. The Grafcet/IEC 1131 standard is not related to data flow graphical programs. Thus the Meyer reference does not teach or suggest the concept of a graphical data flow program as recited in the present claims.

New Claims

Applicant submits new claims 39-82 to more fully and completely claim Applicant’s invention. Applicant submits that these new claims are allowable for at least the reasons given above with respect to claims 1-38.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-18302/JCH.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☒ Information Disclosure Statement
- ☒ Notice of Change of Address

Respectfully submitted,



Jeffrey C. Hood
Reg. No. 35,198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: 2/13/2004

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: 09/466,275
Filed: December 17, 1999
Inventor(s):
Murali Parthasarathy and Omid Sojoodi

Examiner: Chavis, John Q.
Group/Art Unit: 2124
Atty. Dkt. No: 5150-18302

Title: System and Method for Creating
a Graphical Program Which
Invokes Methods and Properties
of Objects

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Commissioner for Patents, Alexandria, VA 22313-1450, on the date indicated below.

Jeffrey C. Hood

2/18/2004
Date

JCH
Signature

NOTICE OF CHANGE OF ADDRESS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Applicant respectfully requests the Commissioner to change the correspondence address for the above identified patent application. Applicant's old address was:

Jeffrey C. Hood
Conley, Rose & Tayon, P.C.
P.O. Box 398
Austin, Texas 78767-0398
(512) 703-1246

Applicant's new address is:

Jeffrey C. Hood
Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, Texas 78767-0398
(512) 853-8820

If there are any questions regarding this matter, please contact me.

Respectfully submitted,

JCH

Jeffrey C. Hood
Reg. No. 35,198

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, Texas 78767-0398
(512) 853-8800

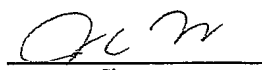
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: 09/466,275
Filed: December 17, 1999
Inventor(s):
Murali Parthasarathy and Omid
Sojoodi

Title: System and Method for
Creating a Graphical
Program Which
Invokes Methods and
Properties of Objects

§ Examiner: Chavis, John Q.
§ Group/Art Unit: 2124
§ Atty. Dkt. No: 5150-18302
§
§
§
§
§
§
§
§
§
§

I hereby certify that this correspondence is being hand delivered to the Commissioner for Patents, Alexandria, VA 22313-1450, on the date indicated below.

 2/18/2004
Signature Date

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
P.O. Box 1450, Alexandria
VA 22313-1450

Sir:

Applicant requests consideration of ☒ the references listed on the attached Form PTO-1449 and/or ☐ the additional information identified below in paragraph 3. A copy of each reference listed on the Form PTO-1449 was previously submitted to or cited by the U.S. Patent and Trademark Office in U.S. Application Serial No. 09/456,892.

1. This Information Disclosure Statement is submitted:
- a. ☐ within 3 months of the filing date of a national application other than a continued prosecution application under § 1.53(d);
☐ within 3 months of the date of entry of the national stage as set forth in § 1.491 in an International application;
☐ before the mailing date of a first Office Action on the merits; or
☐ before the mailing of a first Office Action after the filing of a request for continued examination under § 1.114.
 - b. ☒ after the events of above paragraph 1a and prior to the mailing date of a final Office Action or Notice of Allowance, and thus: ☐ the certification of

- c. ☐ paragraph 2 below is provided, or ☒ a fee of \$180.00 is enclosed below.
after the mailing date of a final Office Action or a Notice of Allowance and prior to payment of the issue fee, and thus: the certification of paragraph 2 below is provided and a fee of \$180.00 is enclosed.

2. It is hereby certified:

- ☐ that each item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the Statement, or
- ☐ that no item of information contained in the Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application or, to the knowledge of the person signing the certification after making reasonable inquiry, was known to any individual designated in § 1.56 (c) more than three months prior to the filing of the Statement.

3. ☐ Consideration of the following additional information (including any co-pending or abandoned U.S. applications, prior uses and/or sales, etc.) is requested:

4. For each non-English language reference listed on the attached Form PTO-1449:

- ☐ reference is made to an English language translation submitted herewith, and/or
- ☐ reference is made to a foreign patent office search report (in the English language) submitted herewith, and/or
- ☐ reference is made to an English language translation of a foreign patent office search report submitted herewith, and/or
- ☐ reference is made to the concise explanation contained in the specification of the present application at page(s) _____, and/or
- ☐ reference is made to the concise explanation set forth below:

5. ☒ Applicant also offers the following comments for the Examiner's consideration:

Please note the following patents: Patent No. 6,064,812, 6,437,805 and 6,102,965 are involved in litigation, Softwire Technology, LLC and Measurement Computing Corporation v. National Instruments Corporation, Civil Action No. 03-10107, in the U.S. District Court for the District of Massachusetts.

Applicant respectfully requests full consideration of each of the references in their entirety.

With respect to reference F1, Applicant respectfully refers the Examiner specifically to pages 2-306 to 2-313.

With respect to reference F3, Applicant respectfully refers the Examiner specifically to pages 2-372 to 2-384.

With respect to reference F4, Applicant respectfully refers the Examiner specifically to chapters 13 and 14.

With respect to reference F5, Applicant respectfully refers the Examiner specifically to page 77 and chapters 13 and 14.

With respect to reference F6, Applicant respectfully refers the Examiner specifically to chapters 13 and 14.

With respect to reference F13, Applicant respectfully refers the Examiner specifically to chapters 1, 6-7, and 12.

With respect to reference F15, Applicant respectfully refers the Examiner specifically to chapters 1, 5-13, 34-37, 61, 76, 69, and 73.

With respect to reference F20, Applicant respectfully refers the Examiner specifically to chapters 1 and 3-5.

With respect to reference F21, Applicant respectfully refers the Examiner specifically to chapters 1 and 4-6.

With respect to reference F22, Applicant respectfully refers the Examiner specifically to chapters 1 and 4-7.

With respect to reference F25A, Applicant respectfully refers the Examiner specifically to chapters 1 and 5.

With respect to reference F29, Applicant respectfully refers the Examiner specifically to chapters 2, 7, and 10 and pages 7-88 to 7-92, pages 10-32 to 10-37, and pages 10-86 to 10-89.

With respect to reference F31, Applicant respectfully refers the Examiner specifically to chapters 8 and 13.

With respect to reference F40, Applicant respectfully refers the Examiner specifically to chapters 1, 7, and 11.

With respect to reference F42, Applicant respectfully refers the Examiner

specifically to chapter 5.

With respect to reference F53, Applicant respectfully refers the Examiner specifically to chapters 1, 3, 5, 10, and 12.

With respect to reference F57, Applicant respectfully refers the Examiner specifically to chapters 16, 20, 21, and 56.

With respect to reference F59, Applicant respectfully refers the Examiner specifically to chapters 1, 2, 4, 6, and 7.

With respect to reference F64, Applicant respectfully refers the Examiner specifically to chapters 2 and 8.

With respect to reference F89, Applicant respectfully refers the Examiner specifically to chapters 1, 2, 9, and 10.

With respect to reference F91, Applicant respectfully refers the Examiner specifically to chapters 1, 3, and 16.

With respect to reference F93, Applicant respectfully refers the Examiner specifically to chapters 2 and 9-14.

6. ☐ Also enclosed is a copy of a foreign search report citing these references.
7. ☐ The listed documents were brought to the attention of the Applicant(s) after payment of the issue fee in the captioned case. The documents were cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this Information Disclosure Statement. Applicant(s) request this Information Disclosure Statement and attached Form PTO-1449 be placed in the file of the captioned application.
8. ☐ Applicant(s) request that the Information Disclosure Statement and attached Form PTO-1449 and references, which are being filed before the grant of the patent and pursuant to 37 C.F.R. § 1.97(i), be placed in the file of the captioned application.

If any required fees are missing, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 501505/5150-18302/JCH.

Respectfully submitted,



Jeffrey C. Hood
Reg. No. 35,198
Attorney for Applicant(s)

MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL PC
P. O. BOX 398
AUSTIN, TEXAS 78767-0398
(512) 853-8800

Date: 2/18/2004

Form PTO-1449 (modified)
List of Patents and Publications
For Applicant's Information
Disclosure Statement
(Use several sheets if necessary)

ATTY. DOCKET NO: 5150-18302

SERIAL NO: 09/466,275

APPLICANT: Parthasarathy et al.

FILING DATE: December 17, 1999

GROUP: 2124

OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)

HP VEE

- | | |
|----|---|
| F1 | "HP VEE Reference" manual, Edition 3, September 1993, 686 pgs. |
| F2 | "How to Use HP VEE" manual, Edition 1, January 1995, 302 pgs. |
| F3 | "HP VEE Reference" manual, Edition 4, January 1995, 771 pgs. |
| F4 | Robert Helsel, "Cutting Your Test Development Time with HP VEE", 1994, 444 pgs. |
| F5 | "HP VEE Advanced Programming Techniques", Edition 4, May 1998, 376 pgs. |
| F6 | Robert Helsel, "Graphical Programming A Tutorial for HP VEE", 1995, 429 pgs. |
| F7 | "HP VEE 3.0 The Most Productive Language for Test & Measurement", 1995, 23 pgs. |
| F8 | "HP VEE 4.0 Visual Programming Language", 1995, 70 pgs. |

IBM VISUALAGE

- | | |
|-----|---|
| F9 | Bitterer et al. "Visual Age and Transaction Processing in a Client/Server Environment", 1996, 369 pgs. |
| F10 | "Developing Distributed Object Applications with VisualAge Smalltalk and SOMobjects, February 1996, 251 pgs. |
| F11 | Stewart McKie, "DBMS - The IBM Enterprise", April 1996, 8 pgs. |
| F12 | Dan Rogers, "VisualAge C++ 3.5 for Windows, October 1996, 6 pgs. |
| F13 | Fang et al., "VisualAge for Smalltalk and SOMobjects, Developing Distributed Object Applications", 1997, 305 pgs. |
| F14 | Marc Carrel-Billiard et al., "Programming with VisualAge for C++ for Windows", 1997, 598 pgs. |
| F15 | "VisualAge for Smalltalk User's Reference, Version 3, Release 0, Third Edition Nov. 1995, 529 pgs. |

LABVIEW

- | | |
|------|---|
| F16 | "LabVIEW for Windows, User Manual", Dec. 1993, 434 pgs. |
| F17 | "LabVIEW Code Interface Reference Manual", Dec. 1993, 249 pgs. |
| F18 | "LabVIEW Code Interface Reference Manual", Sept. 1994, 266 pgs. |
| F19 | "LabVIEW Code Interface Reference Manual", Jan. 1996, 163 pgs. |
| F20 | "LabVIEW Networking Reference Manual", Dec. 1993, 85 pgs. |
| F21 | "LabVIEW Networking Reference Manual", Sept. 1994, 145 pgs. |
| F22 | "LabVIEW Communications VI Reference Manual", Jan. 1996 Edition, 166 pgs. |
| F23 | "LabVIEW for Tutorial", August 1993, 204 pgs. |
| F24 | "LabVIEW for Windows Tutorial", Sept. 1994, 230 pgs. |
| F25 | "LabVIEW Tutorial Manual", January 1996 Edition, 246 pgs. |
| F25A | "LabVIEW Utility Reference Manual" November 1992, 66 pgs. |

Form PTO-1449 (modified)

List of Patents and Publications
For Applicant's Information
Disclosure Statement
(Use several sheets if necessary)

ATTY. DOCKET NO: 5150-18302

SERIAL NO: 09/466,275

APPLICANT: Parthasarathy et al.

FILING DATE: December 17, 1999

GROUP: 2124

Miscellaneous

- F26 "Dr. Dobb's Journal", 1994, 7 pgs. Plus Joseph Firmage, "Novell's AppWare Distributed Bus Extending a Powerful Event Engine Across the Network, 1994, 11 pgs.
- F27 Rijinders et al., "Versatile Visual Programming Environment for Scientific Applications", 1991, 6 pgs.

ICON AUTHOR

- F28 "Software Reviews", Volume 8, September 1993, 16 pgs.
- F29 "Icon Author Reference Manual", Version 7.0, Nov. 1995, 601 pgs.
- F30 "Icon Author User Manual", Version 7.0, Nov. 1995, 382 pgs.
- F31 "Icon Author Getting Started", Version 7.0, Nov. 1995, 205 pgs.

MAX

- F32 "Max and Programming", **NO DATE**, 12 pgs.
- F33 "IRCAM DSP Software for DEC/ALPHA and DEC/MIPS", Aug. 1993, 3 pgs.
- F34 "A Brief History of MAX", 1998, 3 pgs.
- F35 Miller Puckette, "Combining Event and Signal Processing in the MAX Graphical Programming Environment", 1991, 16 pgs.

MS OLE

- F36 Charles Petzold, "Windows 3.1 - Hello to TrueType, OLE, and Easier DDE; Farewell to Real Mode", 12 pgs.
- F37 "MSI Operating System Sensation", downloaded from web 5/9/03, 5 pgs.

Novell

- F38 Jon Udell, "Novell's Campaign - The Vision: a billion connected users and device 2000. The Plan: reinvent its operating systems, p services", Feb. 1995, 15 pgs.

Oberon Prospero

- F40 "Oberon Prospero, User's Guide, Getting Started", Release 1.1, May 1996, 224 pgs.
- F41 Joseph Williams, "DBMS, Prospero 1.1, Sept. 1996, 6 pgs.
- F42 "Application Extensions" Release 1.5, Dec. 1996, 151 pgs.

OOP

- F43 "HP VEE, The Most Productive Language for Test & Measurement", 1995, 25 pgs.

SPW

- F44 "Alta Group Announces CONVERGENCE Simulation Architecture Supporting Full-System, Mixed-Level Verification", Oct. 24, 1995, 3 pgs.
- F45 The Alta Outlook, A custom newsletter for DAC 1996, 34 pgs.
- F46 SPW Product Datasheets and articles, Berkeley Design Technology, Inc., 1995, 47 pgs.
- F47 SPW Product Datasheets, 1994-1996, 181 pgs.
- F47A David Varn, "Riding the Wireless Wave" September 1996, 13 pgs.
- F47B Signal Processing WorkSystem, MATLAB Interface User's Guide incl. notes, October 1995, 83 pgs.

Form PTO-1449 (modified)
List of Patents and Publications
For Applicant's Information
Disclosure Statement
(Use several sheets if necessary)

ATTY. DOCKET NO: 5150-18302

SERIAL NO: 09/466,275

APPLICANT: Parthasarathy et al.

FILING DATE: December 17, 1999

GROUP: 2124

F47C	Shok Indra, "Tool chest continues to grow", EE Times, 12/15/1995, 1 pg.
	SmallTalk
F48	James Carroll, "Smalltalk-based Visual Programming Tools", 1995, 17 pgs.
F49	"HP ORB Plus and Distributed SmallTalk", Sept. 1993, 3 pgs.
	VIPERS
F50	Bernini et al., VIPERS: A Data Flow Visual Programming Environment Based on the Tcl Language", 1994, 3 pgs.
	Digitalk PARTS - VSE
F51	"PARCPLACE-DIGITALK Announces Availability of PARTS(R) for JAA(TM), Ships \$1.2 Million in Product During First Week", July 22, 1996, 2 pgs.
F52	Digitalk List of Books, 1986-1994, 1 pg.
F53	"Smalltalk V, 32-Bit Object-Oriented Programming System Tutorial", version 3.0, Sept. 1994, 203 pgs.
F54	"1995 Digitalk PARTners Catalog", 1994, 40 pgs.
F55	"Distributed Smalltalk Programmer's Reference" Visual Works, 1996, 159 pgs.
F56	"Distributed Smalltalk User's Guide", Visual Works, 1996, 191 pgs.
F57	"Part Reference", VisualSmalltalk Enterprise, 1995, 629 pgs.
F58	"Workbench User's Guide", VisualSmalltalk Enterprise, 1995, 227 pgs.
F59	"User's Guide", VisualSmalltalk Enterprise, 1995, 181 pgs.
F60	"Parts Wrapper for CICS, User's Guide", VisualSmalltalk Enterprise 1996, 112 pgs.
F61	"Tools Reference Manual", VisualSmalltalk Enterprise, 1995, 351 pgs.
F62	"PARTS Workbench Reference Manual", Version 3.0, VisualSmalltalk Enterprise, 1994, 439 pgs
F64	"Tutorial", VisualSmalltalk Enterprise, 1995, 338 pgs.
F65	"Parts Wrapper for CICS/COBOL/EHLLAPI/Relational Databases", 1996, 4 pgs.
F66	Deborah Asbrand, "TMCO Uses Common Sense in Shift to Client/Server", April 1995, 2 pgs.
F67	"Smalltalk Moves into the Fast Lane", April 1995, 1 pg.
F68	"VisualSmalltalk Enterprise", 1995, 4 pgs.
F69	"Parts Wrapper for Relational Databases", May 1995, 1 pg.
F70	Bill Lazar, "Visual Smalltalk Enterprise 3.0", May 1995, 4 pgs.
F71	Brent Whitmore, "Windows-Based Smalltalk Products Go Head to Head", 1995, 6 pgs.
F72	Cate T. Corcoran, "Power Company Juices Up Its Customer Service", Feb. 1995, 2 pgs.
F73	David Linthicum, "Digitalk PARTS", Feb. 1995, 2 pgs.
F74	Rebecca Wirfs-Brock, "How Designs Differ", Dec. 1994, 4 pgs.
F75	Thomas Hoffman, "Capitalizing on Client/Server", Feb. 1995, 2 pgs.
F76	Richard A. Danca, "One Bank's Approach to IT: Standard - But Not Stodgy", Aug. 1994, 4 pgs.

Form PTO-1449 (modified)
List of Patents and Publications
For Applicant's Information
Disclosure Statement
(Use several sheets if necessary)

ATTY. DOCKET NO: 5150-18302

SERIAL NO: 09/466,275

APPLICANT: Parthasarathy et al.

FILING DATE: December 17, 1999

GROUP: 2124

F77	David Linthicum, "Smalltalk: Getting the Message to IS Developers", May 1994, 4 pgs.
F78	Dan Richman, "A Utility Company's New Source of Power", April 1994, 2 pgs.
F79	Brown et al., "Sizing up the Smalltalks", Oct. 1994, 6 pgs.
F80	"The Gospel According to OOP", March 1993, 4 pgs.
F81	"Component-Based Development", April 1993, 4 pgs.
F82	John W. Verity, "Finally, The Buzz is About Smalltalk", April 1993, 2 pgs.
F83	DelRossi et al., PARTS Workbench for Win32 2.0, Elegant and Innovative", Feb. 1994, 4 pgs.
F84	"New Power Source at Utility", Feb. 1995, 1 pg.
F85	Tibbetts et al., "Objects for the Enterprise", 1996, 1 pg.
F86	"HP Distributed Smalltalk User's Guide", 5 th Edition, Oct. 1995, 284 pgs.
	Paragon TNT
F87	Robert C. Waterbury, "Software Trends Lead to Control Forefront", June 1994, 4 pgs.
F88	"Paragon ActiveX Controls Manual", 1997, 280 pgs.
F89	"Paragon TNT Reference Manual", 1996, 969 pgs.
F90	"Paragon TNT OLE Support Guide", 1996, 209 pgs.
F91	"Paragon TNT User's Guide", 1996, 544 pgs.
	ParcPlace-Small Talk
F92	Cunningham et al, "An object-oriented implementation of a graphical-programming system", 1994, 11 pgs.
	ProGraph CPX – Some Pictorius
F93	Steinman et al. "Visual Programming with Prograph CPX", 1995, 418 pgs.
F94	Press Release, "HP ORB Plus and Distributed SmallTalk", 3 pgs.
F95	Raymond Cote, "Prograph CPX: Purely Visual", Jan. 1995, 7 pgs.
F96	"The Power of Prograph CPX", need date, 6 pgs.
F97	Philip T. Cox, "Cross-Platform Application Development with Prograph CPX", need date , 6 pgs.
F98	John Shackelford, "Using Shared Libraries in CPX", need date , 13 pgs.
F99	CPX, ABC Reference, Feb. 1996, 526 pgs.
F100	Prograph CPX for Windows website, 14 pgs.
F101	Visual Programming with Prograph CPX Publication Details from website, 6 pgs.
F102	Prograph CPX for the Macintosh, 3 pgs.
	Pictorius
F103	Rapid Application Development (RAD) with Entrada! Developer and Entrada! Corporate, 7 pgs.
F104	Press releases, 1996, 49 pgs.
F105	The Pictorius Net Servers Mailing List, 13 pgs.
F106	Brett Kottmann, Software Review, Pictorius iNet Developer 2.0, 1996, 7 pgs.
F107	MacWeek Articles, 05/13/1996, 57 pgs.

Form PTO-1449 (modified)
List of Patents and Publications
For Applicant's Information
Disclosure Statement
(Use several sheets if necessary)

ATTY. DOCKET NO: 5150-18302

SERIAL NO: 09/466,275

APPLICANT: Parthasarathy et al.

FILING DATE: December 17, 1999

GROUP: 2124

PSOS

F108 PSOS CORBA Presentation, 1997, 10 pgs.

Miscellaneous

F109 Software, Flowcharter 4.0 by Micrografx, 1995, 3 pgs.

F110 Susan Coote, "Graphical and Iconic Programming Languages for Distributed Process Controls An Object Oriented Approach", IEEE, 1988, 4 pgs.

F111 The History of Visual Basic, Paraphrased from the Visual Basic Programmer's Journal, Feb. 2000, 2 pgs.

EXAMINER:

DATE CONSIDERED:

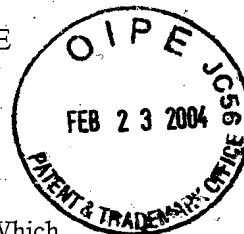
EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the patent owner.

Information Disclosure Statement--PTO 1449 (modified)

K:\NNATL\INST\NATL\INST1\18302\PTO1449.06.doc

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Murali Parthasarathy and Omid Sojoodi
Assignee: National Instruments Corporation
Serial No.: 09/466,275
Filing Date: December 17, 1999
Title: System and Method for Creating a Graphical Program Which
Invokes Methods and Properties of Objects
Atty. Dkt. No.: 5150-18302



The date stamp of the mail room of the U.S. Patent and Trademark Office hereon will acknowledge receipt of the attached: 1) Response to Office Action of January 23, 2004; 2) Notice of Change of Address; 3) Information Disclosure Statement; 4) PTO Form-1449 with References F1 and F111; and 5) Return Postcard.

JCH/jam

Via First Class Mail

Date: 2/18/04

Applicant respectfully requests entry of this Response in the application and requests that the Notice of Abandonment be vacated or withdrawn.

No fees are believed necessary; however, should any fees be required, please appropriately charge those fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-18302/JCH.

Respectfully submitted,



Jeffrey C. Hood
Reg. No. 35,198
ATTORNEY FOR APPLICANTS

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: 7/12/2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Murali Parthasarathy and Omid Sojoodi
Assignee: National Instruments Corporation
Serial No.: 09/466,275
Filing Date: December 17, 1999
Title: System and Method for Creating a Graphical Program Which
Invokes Methods and Properties of Objects

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Murali Parthasarathy and Omid Sojoodi
Assignee: National Instruments Corporation
Serial No.: 09/466,275
Filing Date: December 17, 1999
Title: System and Method for Creating a Graphical Program Which
Invokes Methods and Properties of Objects
Atty. Dkt. No.: 5150-18302



The date stamp of the mail room of the U.S. Patent and Trademark Office hereon will acknowledge receipt of the attached: 1) Petition to Withdraw Holding of Abandonment Under CFR 1.181(a); 2) copy of Response to Notice of Non-Compliant Amendment Mailed January 23, 2004-Response to Office Action of September 10, 2002; 3) a copy of the return receipt postcard stamped as received by the U.S. Patent Office on February 23, 2004; and 4) Return Postcard.

JCH/jam

Via First Class Mail

Date: 8-12-05

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.:	09/466,275	§	
Filed:	December 17, 1999	§	Examiner: Chavis, John Q.
Inventor(s):		§	Group/Art Unit: 2191
Murali Parthasarathy and Omid		§	Atty. Dkt. No: 5150-18302
Sojoodi		§	
		§	
Title:	System and Method for	§	
	Creating a Graphical	§	
	Program Which	§	
	Invokes Methods and	§	
	Properties of Objects	§	
		§	
		§	

STATUS INQUIRY

Dear Sir or Madam:

On August 15, 2005, Applicant's undersigned attorney filed a Petition to Withdraw Holding of Abandonment Under CFR 1.181(a) regarding the above-referenced patent application with the U.S. Patent and Trademark Office. Applicant has not heard from the Patent Office since then. Applicant hereby respectfully requests the Patent Office to notify Applicant as to the status of the matter.

Should any fees be required, the Commissioner of hereby authorized to charge such fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-18302/JCH.

Respectfully submitted,

/Jeffrey C. Hood/

Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: January 29, 2007 JCH/JAM

Electronic Acknowledgement Receipt

EFS ID:	1472169
Application Number:	09466275
International Application Number:	
Confirmation Number:	9971
Title of Invention:	SYSTEM AND METHOD FOR CREATING A GRAPHICAL PROGRAM WHICH INVOKES METHODS AND PROPERTIES OF OBJECTS
First Named Inventor/Applicant Name:	MURALI PARTHASARATHY
Customer Number:	35690
Filer:	Jeffrey C Hood/Jeannie Miranda
Filer Authorized By:	Jeffrey C Hood
Attorney Docket Number:	5150-18302
Receipt Date:	29-JAN-2007
Filing Date:	17-DEC-1999
Time Stamp:	14:01:17
Application Type:	Utility

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)	Multi Part /.zip	Pages (if appl.)
1	Request for status of Application	5150-18302_Status_Inquiry_Petition.pdf	74092	no	1

Warnings:

Information:**Total Files Size (in bytes):**

74092

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.